

学习资料:

- [LaTeX笔记（八）——数学建模专题](#)
- [如何用Markdown写论文？](#)
- [LaTeX排版札记](#)
- [LaTeX排版札记：part 2—速查手册、导言区、扉页和公式](#)

- [论文格式细节整理汇总](#)

- <https://www.zhihu.com/question/66167182/answer/239342568>

- 只要遇到问题，Google 一下，后面加一个 `site:http://stackoverflow.com`，能解决你 99.999 的疑问，还能顺便把英文练好。
- 新手不要自己造轮子，你 99 的需求都有宏包支持，你只需要找到是哪个宏包，并且看懂手册就行了。
- 查找手册可以在命令行里用 `texdoc` + 宏包名字，执行起来有点小慢，我喜欢在 Everything 里搜索宏包名字 + PDF。
- Git 使用，那真是想删哪就删哪，大不了再退回来。而且，晚上回家前 push 到网上，回家可以继续写。多人合作一篇文章也不错。
- 写中文文档一定要用 `ctex` 宏包，不要用 `CJK` 或者 `xeCJK` 宏包。
- `tabu` 比 `tabular` 好用太多，`cleveref` 提供的 `\cref` 比默认的 `\ref` 好用太多。
- 关于 label 的命名，我建议 `perfix:caption`，perfix 就是前缀，比如 `fig`、`tab`、`lst`、`eqn`、等等，一看就知道这个 label 是什么类型。前缀加 `caption` 的组合，可以很容易的找到需要引用的 label，而不用到处来回翻找。
- 数学公式，要规范，花点时间读一下 [ChinaTEX MathFAQ](#)。
- 不要忽视 Warning，最好弄明白每个 Warning 是什么意思，怎么产生的，会对文档造成什么影响，如何消除。实在觉得 Warning 碍眼的话，可以用 `silence` 宏包屏蔽掉。
- 写特别大的文档，要将整个文档分成若干部分，用 `\input` 或者 `\include` 加到主文件中。
- 论文实验数据绘图可以考虑用 `pgfplots`，实验代码只需要把结果保存成文本格式的数据就行。有两点好处：第一，图的风格与正文一致，强迫症福音；第二，修改实验不需要改图，只需要重新跑一下实验，然后重新用 LaTeX 编译一下就行了。
- 如果你是强迫症晚期的话，其他插图可以考虑用 `tikz` 宏包完成。
- LaTeX 表格对新手不是特别友好，可以考虑用这个 <https://www.tablesgenerator.com/>。
- 说到表格，不管你用什么编辑器，一定要用支持列模式的编辑器，一定要用等宽字体，千万别用 Times New Roman 等非等宽字体的，画表格想死。
- 最后推荐几个我经常用的宏包吧。
 - `tcolorbox`，简直是黑魔法的存在；
 - `minted/listings`，排版代码需要他们；
 - `enumitem`，列表环境想怎么调就怎么调；
 - `xparse`，提供 `\NewDocumentCommand` 等命令，方便定义形参复杂的命令。

0. 使用 LaTeX 有什么比较好的编写技巧或习惯？

Detexify

识别手写的 symbol 然后给出相应的 Latex 代码并且给出需要 include 的 package，识别的准确度很高。

制作表格

在线制作 Latex 表格的网站神器：**LaTeX Table Generator**！真正的所见即所得，支持代码与表格之间的无缝转换！更支持与Excel的复制粘贴，非常好用！

除去 Latex 之外，网站还支持 Markdown, HTML 等多种格式的表格生成。再也不用花时间做表格了！（当然，如果需要特别精细的调整，还是需要结合自己的需要灵活用LaTex代码调整。这个工具只是辅助，但是基本上能解决很多的大问题的。）

地址在这里：<http://www.tablesgenerator.com>

在线LaTex协作编译器 -- 合作者的利器 Overleaf

网址：<https://www.overleaf.com/>

自定义一些命令

例如：`\def\R{\mathbb{R}}`

这样你敲 `\R` 就等价于 敲 `\mathbb{R}` 了。

`.bib` 文档

把所有 citation 都放在一个叫 `research.bib` 的文档里，然后代码的最后加上以下两行：

```
1. \bibliographystyle{unsrt}
2. \bibliography{research}
```

bib文档中一个引用是长这样的：

```
1. @ARTICLE{potts,
2.   author = {{Potts}, R.~B. and {Domb}, C.},
3.   title = {Some generalized order-disorder transformations},
4.   journal = {Proceedings of the Cambridge Philosophical Society},
5.   year = 1952,
6.   volume = 48,
7.   pages = {106--109},
8. }
```

这样的好处是，容易管理 citation，方便重复利用。

以下我均是在 **TeXstudio** 软件环境下编译的！

TeXstudio 有很丰富的向导操作，比如在 `向导(w)` 中有插入图片按钮，十分方便。

- [TeXstudio : User manual](#)

1. 基本结构及操作

构成一篇文档最基本的三个语句是：

```
1. \documentclass{book} %模板类型为book
2.
3. \title{...} %添加标题
4.
5. \author{...} %添加作者
6.
7. \date{} %LaTeX会自动生成日期，如果不需要就加这一步将日期去掉
8.
9. \begin{document} %开始正文
10. \maketitle %制作封面
11.
12. \tableofcontents %加入目录，包括页码（非必需）
13.
14. \mainmatter %如果希望页码是从正文部分开始就加上这句命令（非必需）
15.
16. %下面是几个主要的层次结构；
17.
18. \part{...} %第一部分
19. \chapter{...} %第一章
20. \section{...} %第一节
21. \subsection{...} %第一小节
22.
23. %可以重复排列添加上面的层次结构。
24.
25. \part{...} %第二部分
26. \chapter{...} %第二章
27. \section{...} %第二节
28. \subsection{...} %第二小节
```

- 注意一件事情：如果输入 `\tableofcontents` 但是pdf格式中却什么都没有出现，那么需要再重复几次 L 与 dvi 转 pdf 过程。

2. 列表和表格环境

2.1 列表

LaTeX 的列表分为**无序列表**和**自定义列表**。

无序列表

```
1. \documentclass[10pt,a4paper]{article}
2. \usepackage[utf8]{inputenc}
3. \usepackage{amsmath}
4. \usepackage{amsfonts}
5. \usepackage{amssymb}
6. \usepackage{graphicx}
7. \begin{document}
8.
9.     \begin{itemize}
10.    \item This is the first row.
11.    \item This is the first row.
12.    \item This is the first row.
13. \end{itemize}
14.
15. \end{document}
```

自定义列表

需要使用宏包，命令为：

```
1. \usepackage{...} %...为宏包名字
```

简单来说，这个过程就像是 Python 里面的 import 和 C 语言里面的 include。

```
1.
2. \usepackage{enumerate}
3.
4. \begin{enumerate}[A.] %这里我们将项目符号自定义为大写字母., 也就是说自定义过程需要将内容加在后面的[]中。
5.     \item hello
6.     \item hello,suri
7.     \item hello
8. \end{enumerate}
```

同样可以把 A. 换成任何自定义的符号。另外也可以在 [] 中对项目符号的格式进行修改。比如：

```
1.
2. \begin{enumerate}[\beseries A.] %对项目符号加粗
3. \begin{enumerate}[\sffamily A.] %无衬线字体
```

2.2 表格

表格基本操作

插入表格的命令为编辑器上方的 **向导** → **表格**，点击之后确认行数和列数（这里是三行三列），将会出现：

```
1. \begin{tabular}{| *|*|*| } %这里的内容是为了确认对齐情况
2. \hline
3. % after \\\: \hline or \cline{col1-col2} \cline{col3-col4} ...
4. * & * & * \\ %将内容输入在*的位置
5. * & * & * \\
6. * & * & * \\
7. \hline
8. \end{tabular}
```

- 表格内容对齐的方式有三种，分别是**居中对齐**，**靠左对齐**，**靠右对齐**。三种格式反映在表格第一行，代表符号分别为 **c**，**l**，**r**（英文首字母）

例如，如果我们需要将这份表格内容全部居中对齐，则输入：

```
1. \begin{tabular}{|c|c|c|}
```

- 这里的竖线代表了输出之后的表格中会出现竖的表格线，如果不需要可以去掉，即为：

```
1. \begin{tabular}{ccc}
2.
3. \hline %添加横线，单独一行
```

- `\cline{2-3}` 表示在是在上一行的第二个第三个单元格下面添加横线。
- `\\` 用在每一行的后面，表示表格中的换行
- `\multicolumn {number} {...}{...}` 合并列，其中 `number` 为要合并的列的数量，第一个 ... 为对齐方式，第二个为合并的内容。只对该行有效。
- `\multirow` 合并行，需要导入该宏包
- `&` 是分隔符

添加表头

- `table` :
这里的 `caption` 是标题，`label` 主要在添加超链接时使用，`centering` 表示所有内容居中对齐。

```
1. \begin{table}
2. \centering
3. \begin{tabular}{cccc}
4. \hline
5. eshi&sd&dfefe&sdsd\\
6. \hline
7. 12&343&423&312\\
8. 123&32&132&312\\
9. 23&434&231&23\\
10. \hline
11. \end{tabular}
12. \caption{this is my table}
```

```
13. \end{table}
```

- 注：如果表格太长无法在一页的话，需要将 `table` 换成 `longtable`，并需要使用宏包。

3 特殊字符

- `%` 后面为注释；输入正规的 `%` 需要在 `%` 前面加 `\`
- `$` 在数学状态下使用，数学公式有行间公式和行内公式两部分组成，行间公式需要在两个 `$$` 之间进行，居中对齐。一个 `$` 是生成行内公式；正规的 `$` 也需要 `\`
- `{}` 属于内部命令，需要在前段和后端加 `\`
- `\` 在 latex 有专门的命令为 `backslash`

处理中文时应该用 ctex 宏包

CTeX 和 TeX Live 套件都包含 ctex 处理中文的包，可以看下 [ctex 的文档](#)

- 全中文文档建议使用 cTeX 文档类：`ctexart`、`ctexrep`、`ctexbook`、`ctexbeamer`
- ```
\documentclass{ctexart}
```

或者直接使用

```
1.
2. \documentclass{ctex}
```

## 4 公式编辑

latex 公式有两种，分别为**行内公式**和**行间公式**（行间公式有两种插入方式）：

```
1.
2. $a^2+b^2=c^2$ %行内公式
3. $$a^2+b^2=c^2$$ %行间公式
4. \[a^2+b^2=c^2\] %行间公式
```

### 4.1 对公式编号

```
1.
2. \begin{equation}\label{} %label仍然表示标签，只有引用时用得到，非必需
3.
4. \end{equation}
```

示例：

```
1. \begin{equation}
2. \frac{{- b \pm \sqrt{{b^2} - 4ac} }}{{2a}}
3. \end{equation}
```

编号时根据章节来的，若程序中没有层次结构，所以显示为 (1)，这个是可以人为控制的，想要按照哪一部分来编号，可以在前面加代码：

```
1. \numberwithin{equation}{subection} %按照subection对公式编号
```

而有的时候，如果出现两个重要程度等同的公式需要用ab进行编号，那么需要使用的命令是：

```
1. \begin{subequations}
2. *%公式
3. \end{subequations} %使用ab进行编号
```

需要注意的是，使用 subequations 命令需要使用到宏包 amsmath，需要在开头导入宏包。

```
1. \documentclass{book}
2. \usepackage{amsmath} %导入宏包
```

使用方法举例如下：

```
1. \begin{subequations}
2. \begin{equation}
3. \frac{{- b \pm \sqrt{{b^2} - 4ac} }}{{2a}}
4. \end{equation}
5.
6. \begin{equation}
7. \frac{{- b \pm \sqrt{{b^2} - 4ac} }}{{2a}}
8. \end{equation}
9. \end{subequations}
```

## 4.2 公式格式修改

为了起到强调的作用，我们可以给公式加一个框。加框的命令为：

```
1. \boxed{}
```

boxed 命令必须是在 equation 环境之下，举例用法如下：

```
1. \begin{equation}\boxed{
2. \frac{{- b \pm \sqrt{{b^2} - 4ac} }}{{2a}}}
3. \end{equation}
```

展示如下：

%%latex

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(1)

### 4.3 通过自定义简化函数

在 latex 中有些命令非常长，在输入公式的时候会带来很多的不便，为了简化这些命令，我们可以使用 `newcommand`，这个命令大致相当于 C语言 中的宏替换，用法如下：

1.

2.

```
\newcommand{\fc}{\frac}
```

%这个命令使用在文章的开头，表示用 fc 来表示 frac，同时 frac 函数仍然可以继续使用。

之后在后面需要使用 `frac` 函数时，只需要输入 `fc` 即可，我们可以进行对比：

1.

2.

3.

```
$$\frac{{ - b \pm \sqrt{{b^2} - 4ac} }}{{2a}}$$
$$\fc{{ - b \pm \sqrt{{b^2} - 4ac} }}{{2a}}$$
```

但是需要注意的是，自定义的名称不能够与 latex 内置的命令冲突。

## 5 图片的插入

### TeXstudio 可以直接插入图片

使用 latex 插入图片时基本的操作有两个：

- 第一个是需要使用宏包：

1.

```
\usepackage{graphicx}
```

%在导言区导入宏包

• 第二个为格式

1.

```
\includegraphics{*.eps}
```

/\*代表了文件名

下面以不同格式的图片为例进行介绍：

- 插入 `eps` 格式图片

EPS是 Encapsulated Post Script 的缩写。EPS 格式是 Illustrator CS5 和 Photoshop CS5 之间可交换的文件格式。EPS 文件是目前桌面印刷系统普遍使用的通用交换格式当中的一种综合格式。——百度百科

示例：



```

1. \begin{figure}
2. \centering
3. \includegraphics[scale=0.2]{hellohellotest.eps} %这里 scale 控制图片的大小为原图片的 0.2倍, 备注
 : 其实可以不加 .eps
4. \caption{This is a Poster} %图片标题
5. \end{figure}

```

控制图片输出格式的几个常见的具体命令：

- `width` =\* %宽度
- `height` =\* %高度 高度和宽度必须标明明确的单位，比如厘米（cm）或者英寸（in）
- `scale` =\* %倍数
- `angle` =\* %顺时针旋转角度

## 6 数学公式排版进阶

事实上，`$$` 因为会产生一些不良的间距，缺少错误检查，并且不能正确处理 `fleqn` 等文档选项，所以不推荐。一般的上下标都是放在数学符号的右上，右下方，如果需要将它们写在正下，正上方（比如写 `\max`，写  $\Sigma$  什么的），我们可以使用 `\limits`。

```

1. $\sum_{i=1}^n \quad \quad \quad %\quad表示拉开一段距离
2. $\sum\limits_{i=1}^n$

```

$$\sum_{i=1}^n$$

想加入多行的上下标？可以使用 `\substack` 命令，举个例子：

```

1. $\sum\limits_{\substack{i=1\\j=1}}^n$ %注意\substack使用的位置

```

$$\sum_{i=1 \atop j=1}^n$$

### 6.1 定界符

嵌套多了式子会变得非常复杂，也就会变得越来越大！可是这个时候如果你使用括号你会发现，它的大小并没有什么变化，这就显得非常的 low，影响美观，因此我们会在括号外加一个 `left` 或者是 `right` 进行大小的控制。

### 6.2 标准的LaTeX提供的数学字体有以下几种

```

1. \[

```

```

2. \mathit{ABCDE}\]
3. \[
4. \mathrm{ABCDE}\]
5. \[
6. \mathbf{ABCDE}\]
7. \[
8. \mathsf{ABCDE}\]
9. \[
10. \mathtt{ABCDE}\]

```

```

1. %%latex
2. \[
3. \mathit{ABCDE}\]
4. \[
5. \mathrm{ABCDE}\]
6. \[
7. \mathbf{ABCDE}\]
8. \[
9. \mathsf{ABCDE}\]
10. \[
11. \mathtt{ABCDE}\]

```

```

[
\mathit{ABCDE}]
[
\mathrm{ABCDE}\]
[
\mathbf{ABCDE}\]
[
\mathsf{ABCDE}\]
[
\mathtt{ABCDE}\]

```

## 数学公式对齐

对齐功能一般使用align环境：

注意：

- 每一行都有一个换行 `\\` 和一个 `&` 符号，每一行按照 `&` 对齐，即对齐之后的所有内容。
- 但是有一点不太好就是每一行之后都会显示公式的编号，这又显得不太好。这个时候只要在每一行的最后（`\\` 的之前）加一个命令 `\notag` 即可去掉本行公式的标号。当然，也可以使用 `align*` 环境，这种环境可以直接消去所有编号。

```

1. %%latex
2. \begin{align}
3. &\lim\limits_{x\to 1}\left(\frac{1}{1-x}-\frac{3}{1-x^3}\right)\\
4. &= \lim\limits_{x\to 1}\left(\frac{x^2+x-2}{1-x^3}\right) \quad \notag

```

```

5. =& \lim\limits_{x\to 1}\frac{(x+2)(x-1)}{(1-x)(x^2+x+1)}\\
6. =& \lim\limits_{x\to 1}\frac{-(x+2)}{x^2+x+1}\\
7. =& -1
8. \end{align}

```

$$\lim_{x \rightarrow 1} \left( \frac{1}{1-x} - \frac{3}{1-x^3} \right) \quad (2)$$

$$= \lim_{x \rightarrow 1} \left( \frac{x^2 + x - 2}{1 - x^3} \right) \quad (3)$$

$$= \lim_{x \rightarrow 1} \frac{(x+2)(x-1)}{(1-x)(x^2+x+1)} \quad (4)$$

$$= \lim_{x \rightarrow 1} \frac{-(x+2)}{x^2+x+1} \quad (5)$$

$$= -1 \quad (6)$$